

FIG. 1A

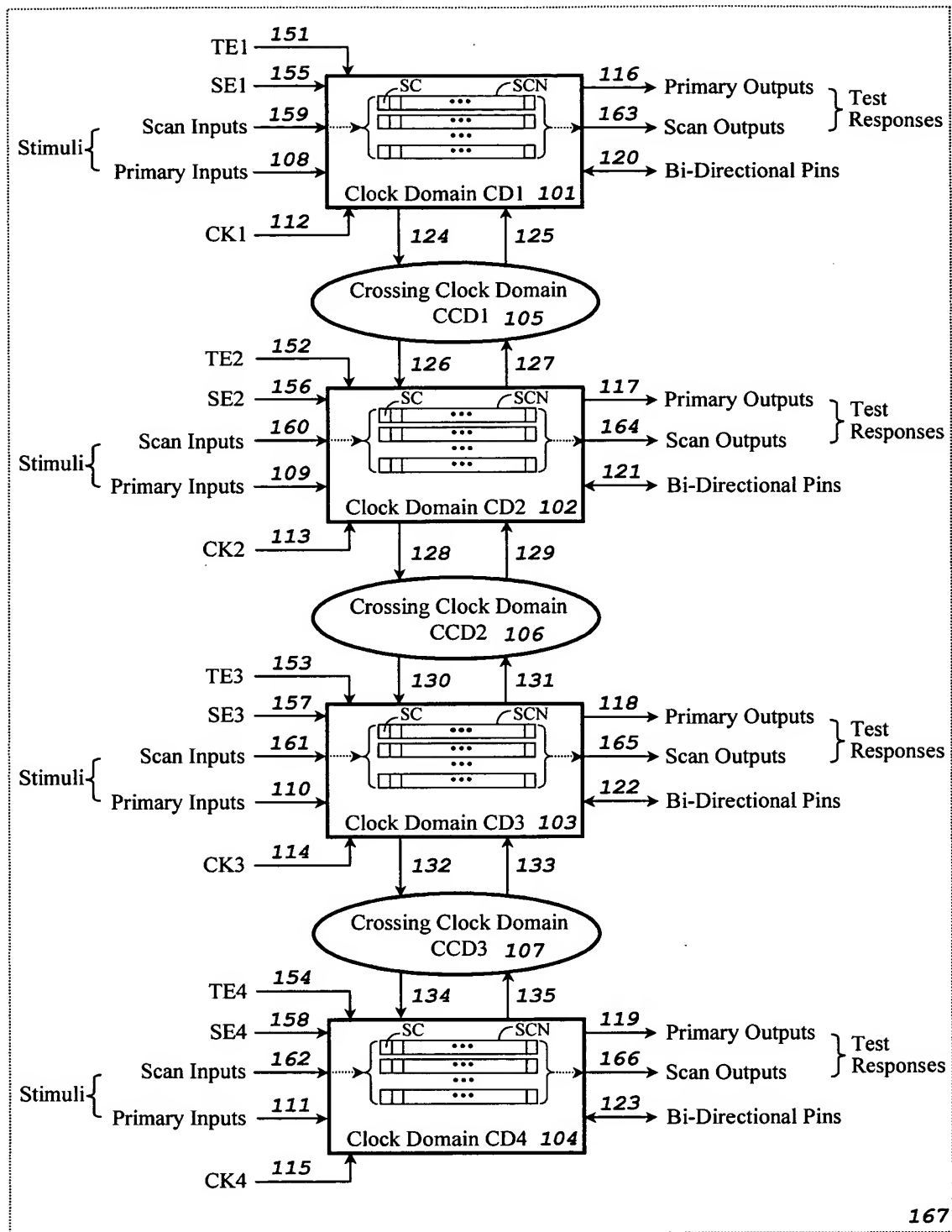


FIG. 1B

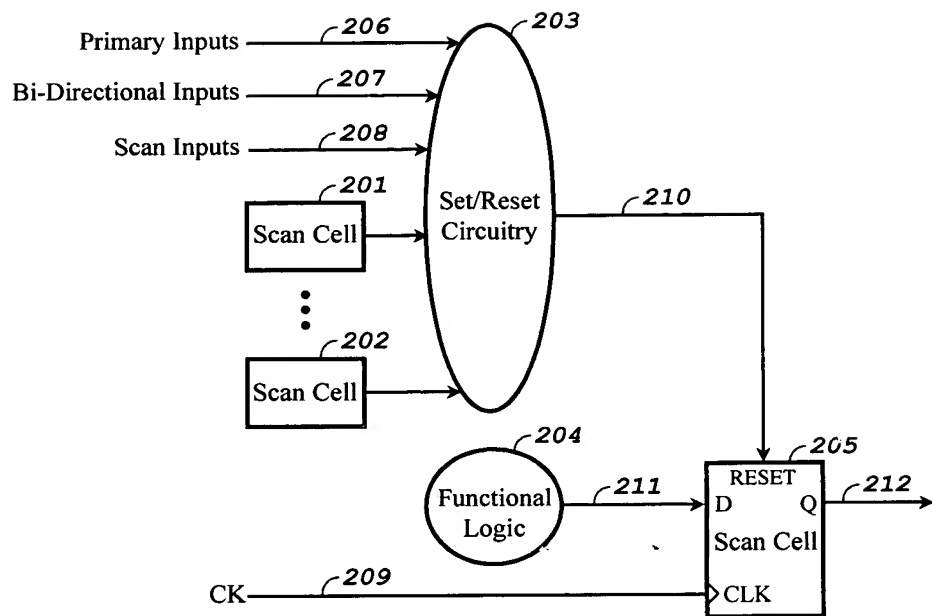
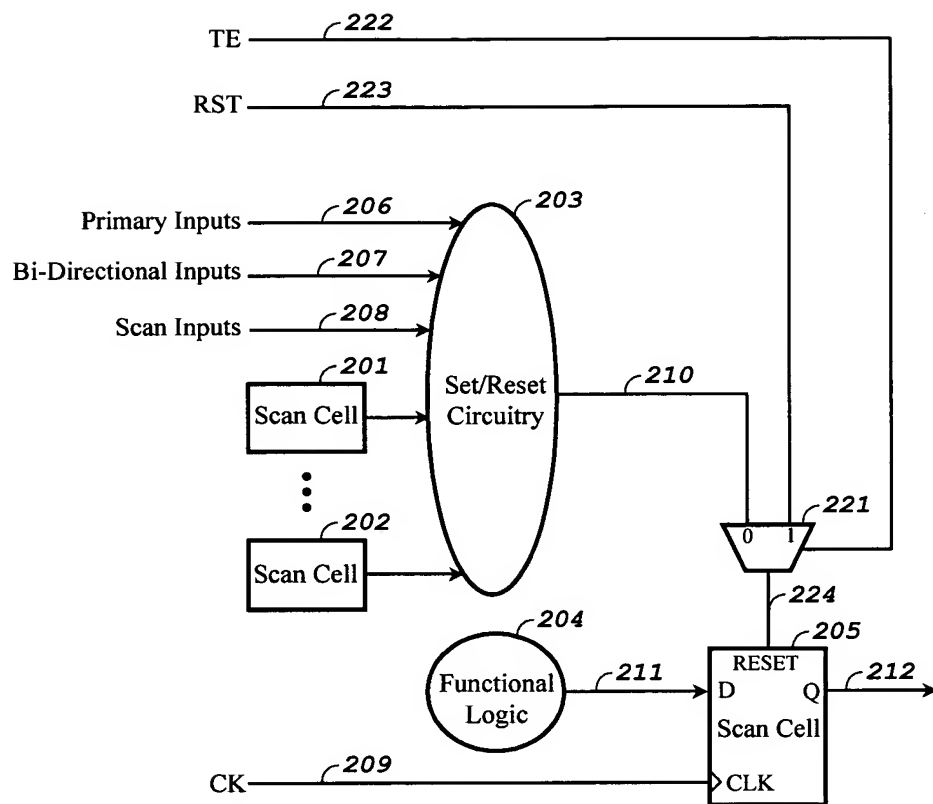
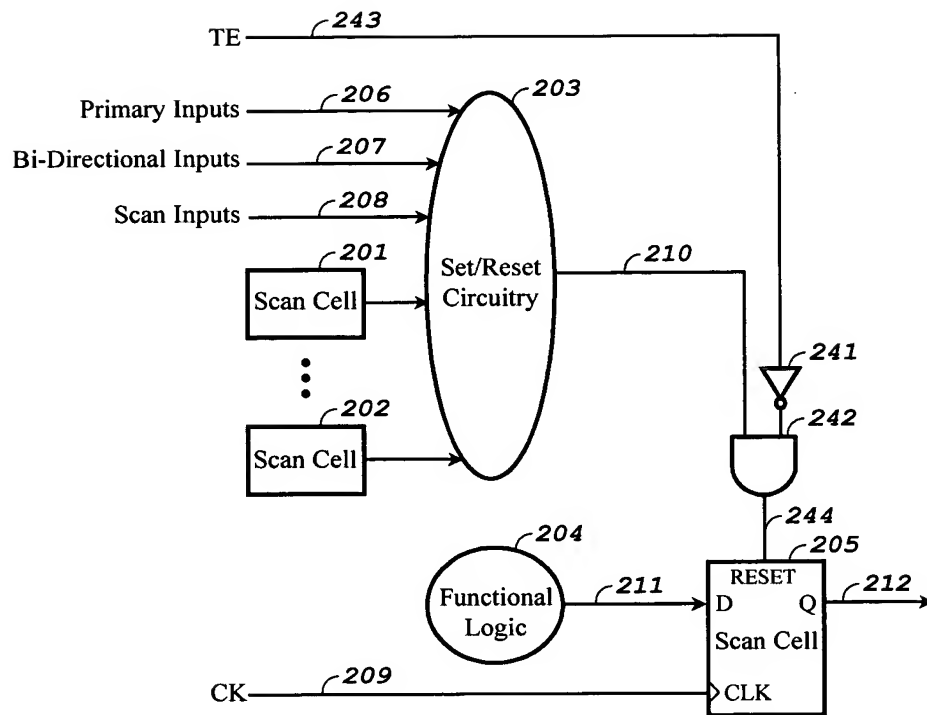


FIG. 2A



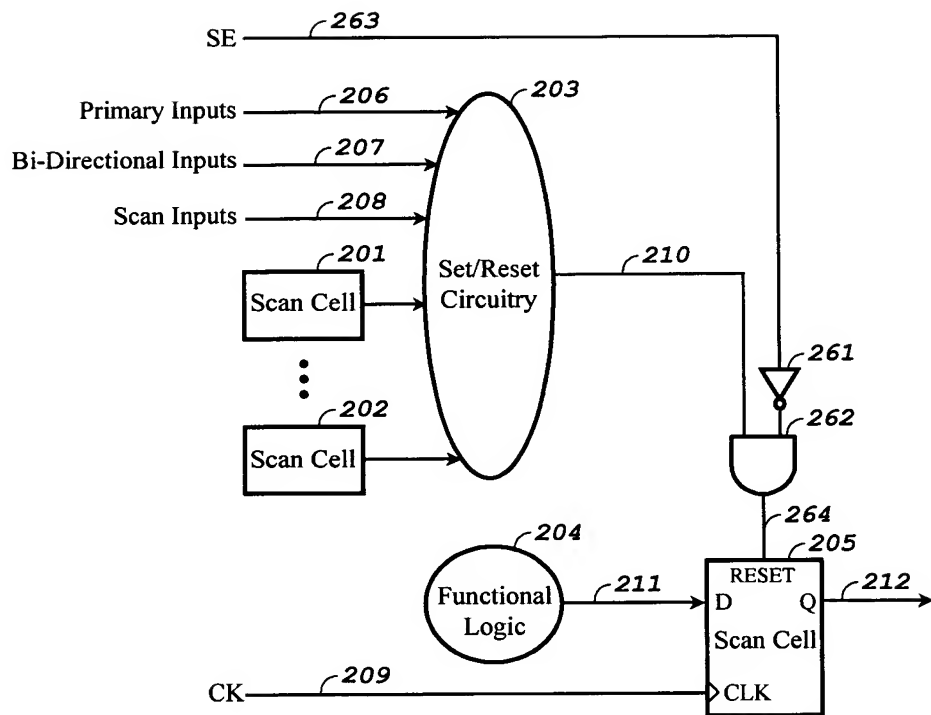
PRIOR-ART #1

FIG. 2B



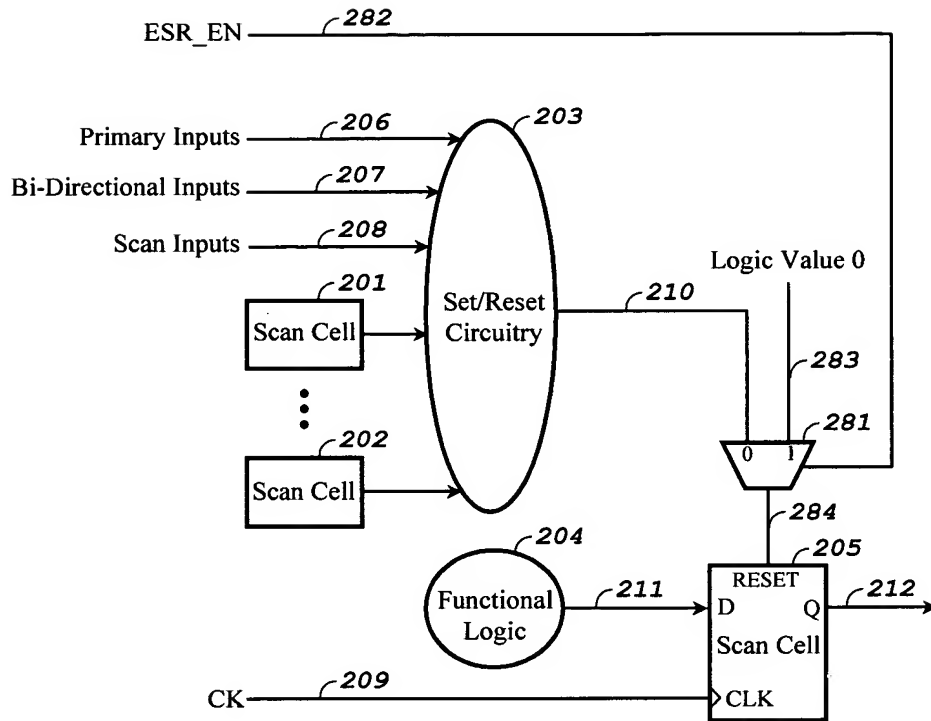
PRIOR-ART #2

FIG. 2C



PRIOR-ART #3

FIG. 2D



PRIOR-ART #4

FIG. 2E

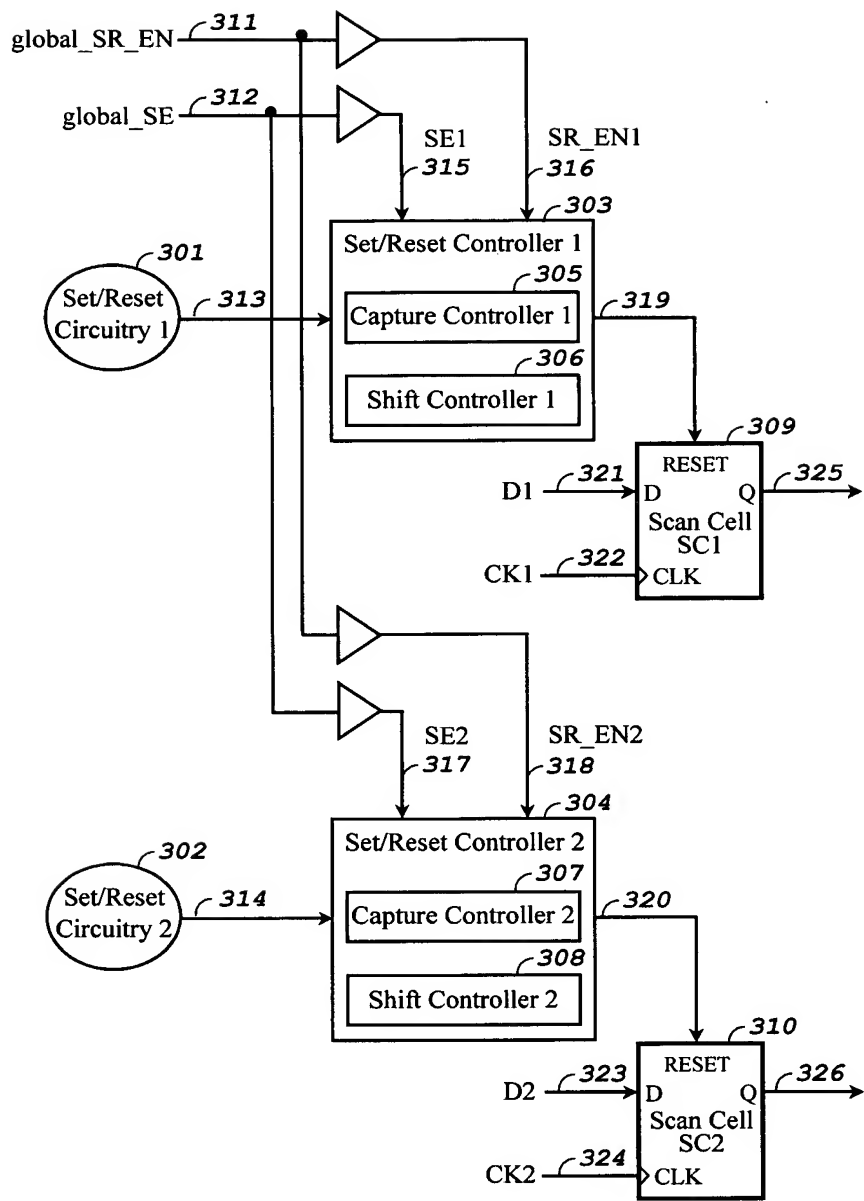


FIG. 3A



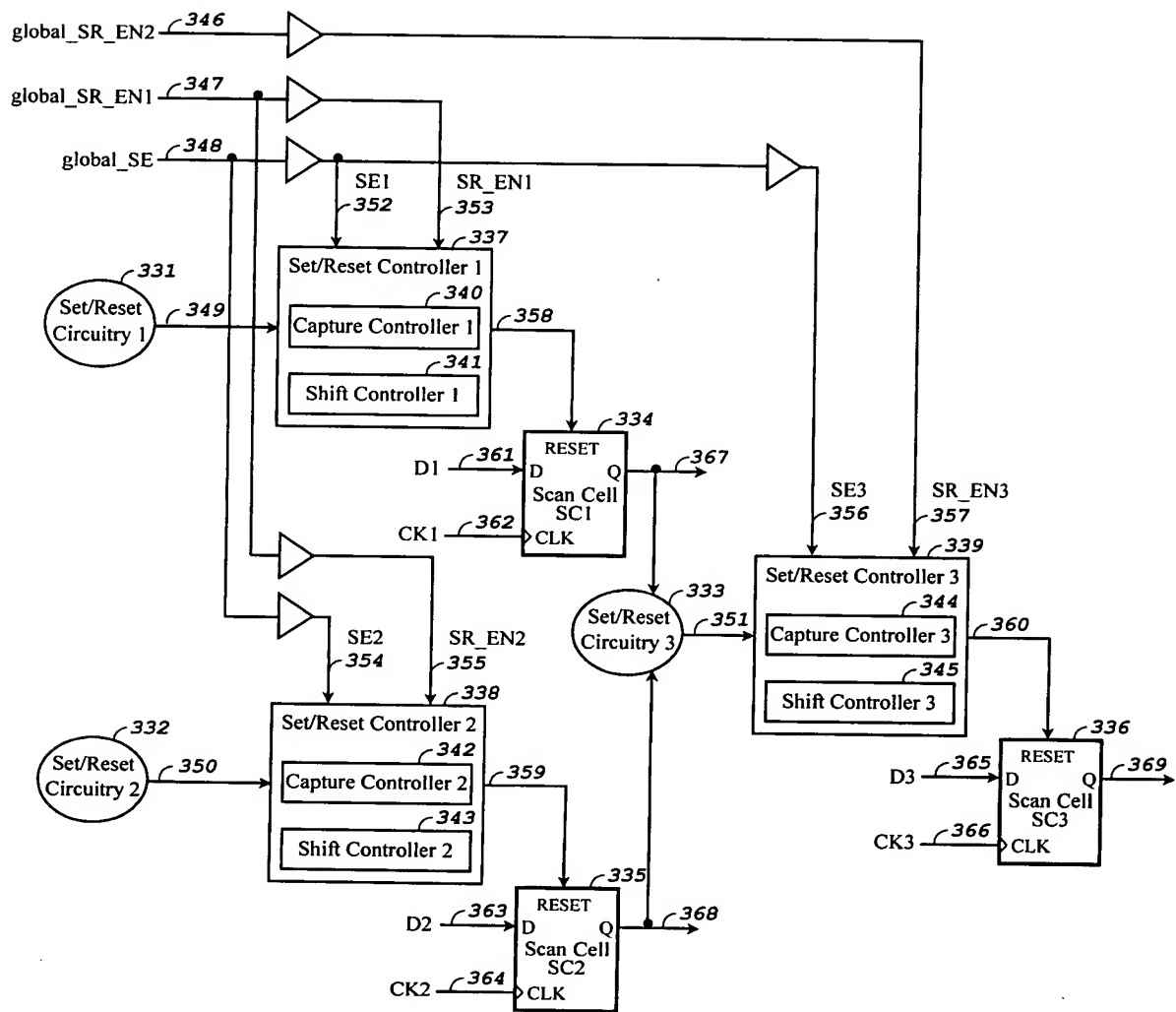


FIG. 3B

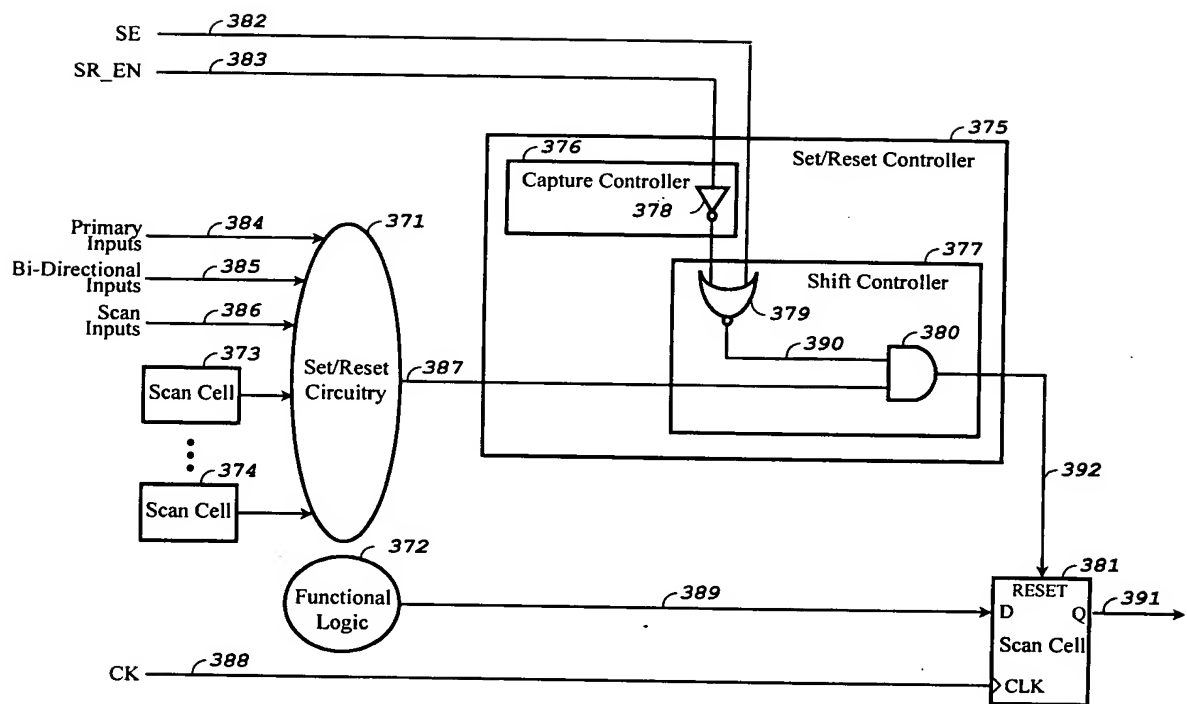


FIG. 3C

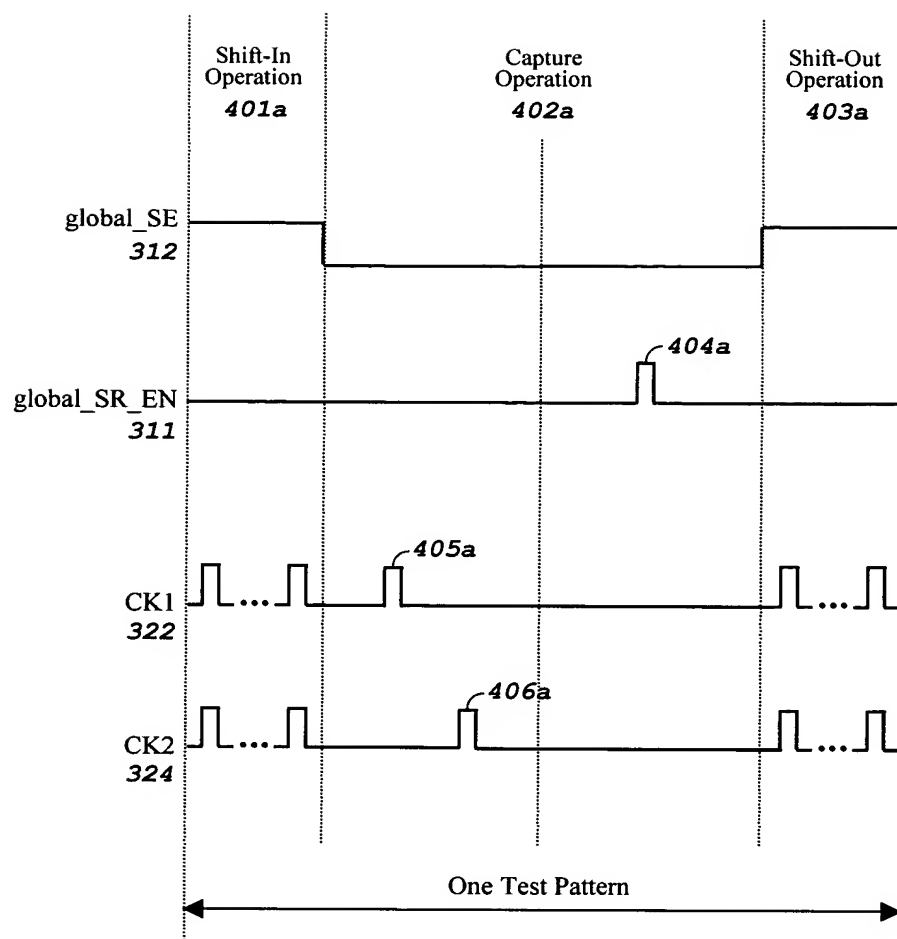


FIG. 4A

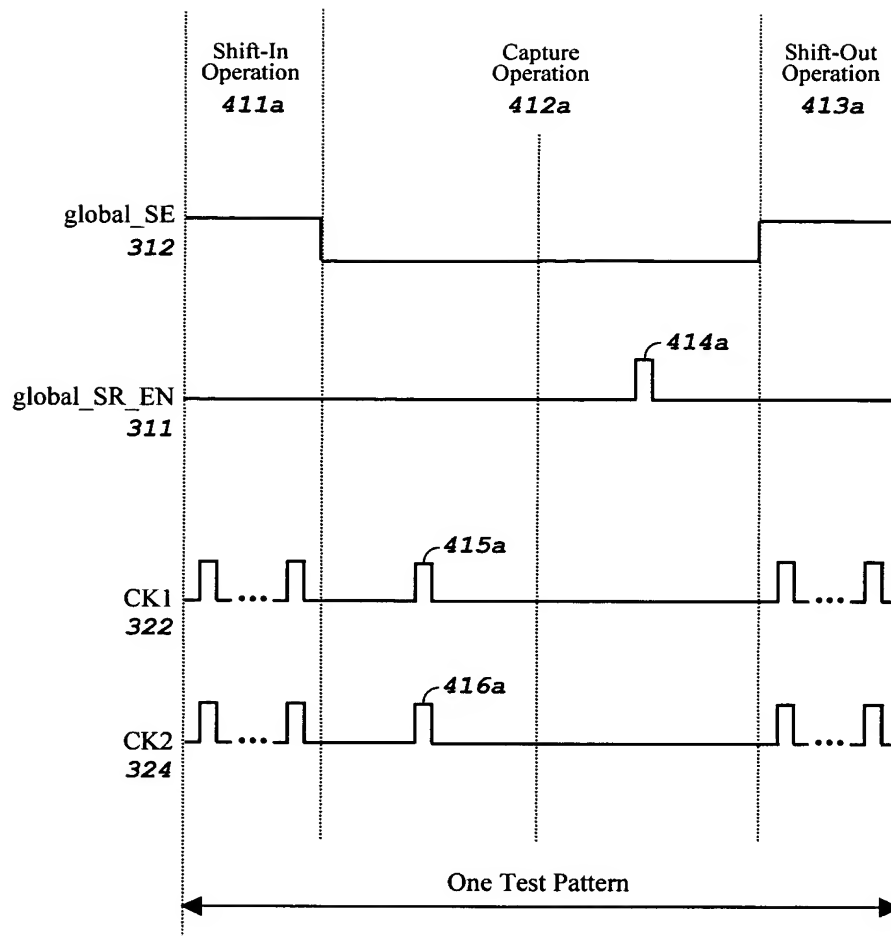


FIG. 4B

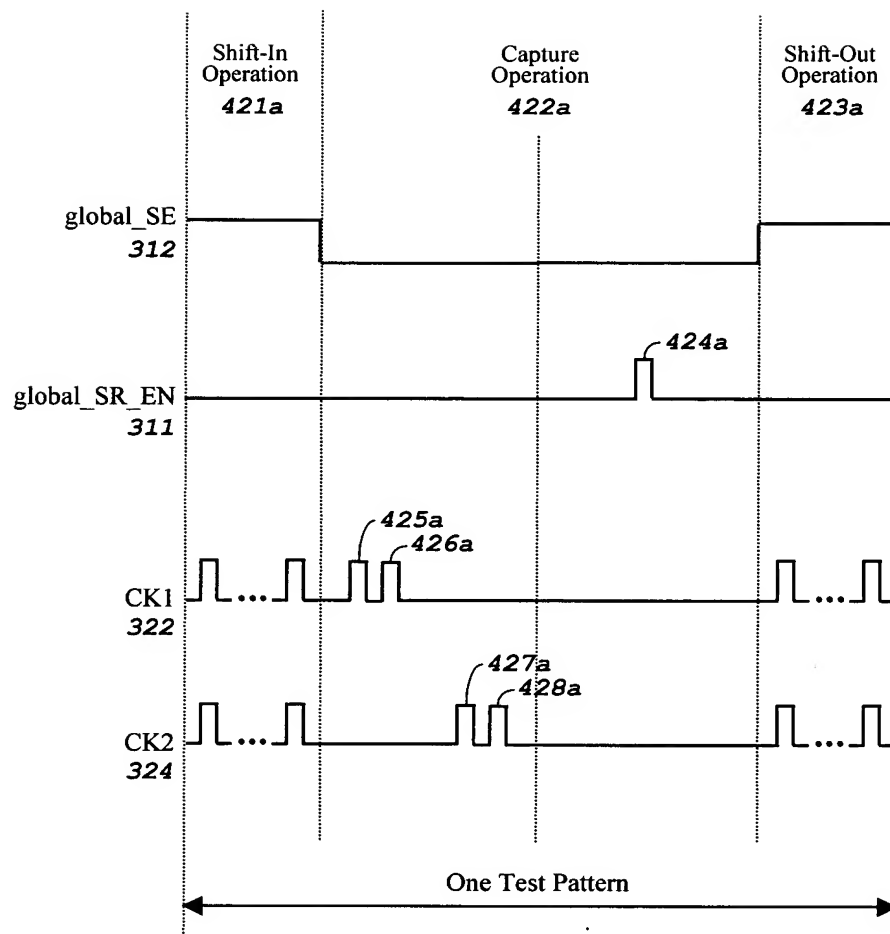


FIG. 4C

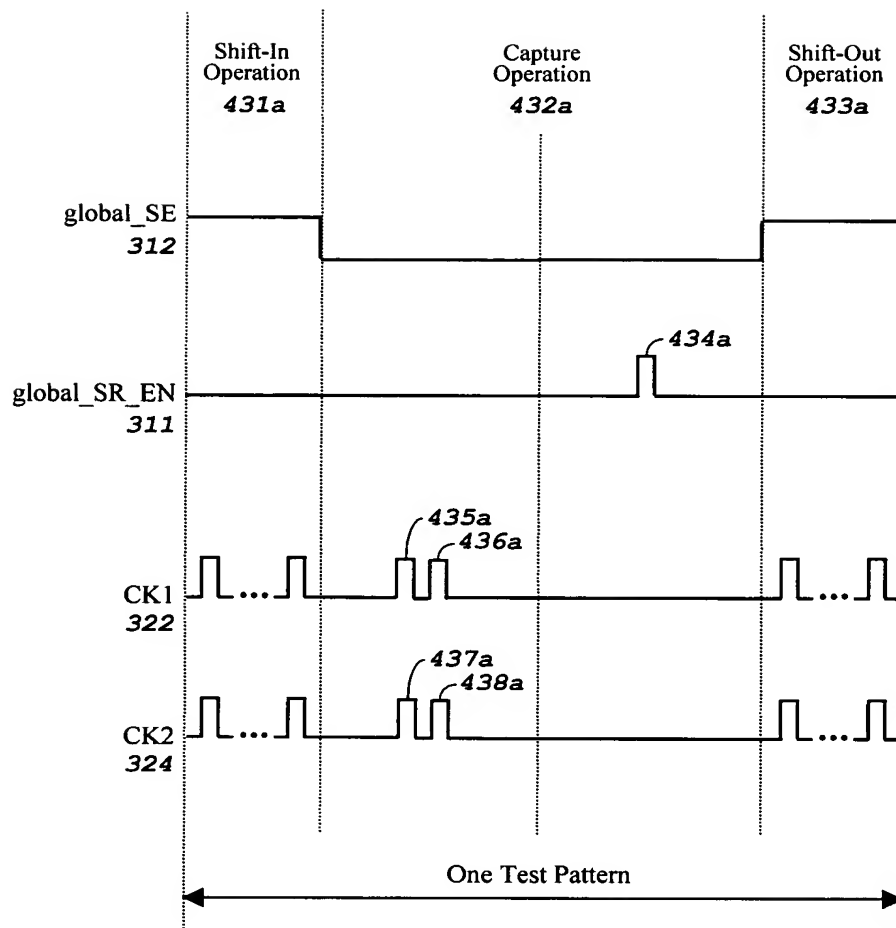


FIG. 4D

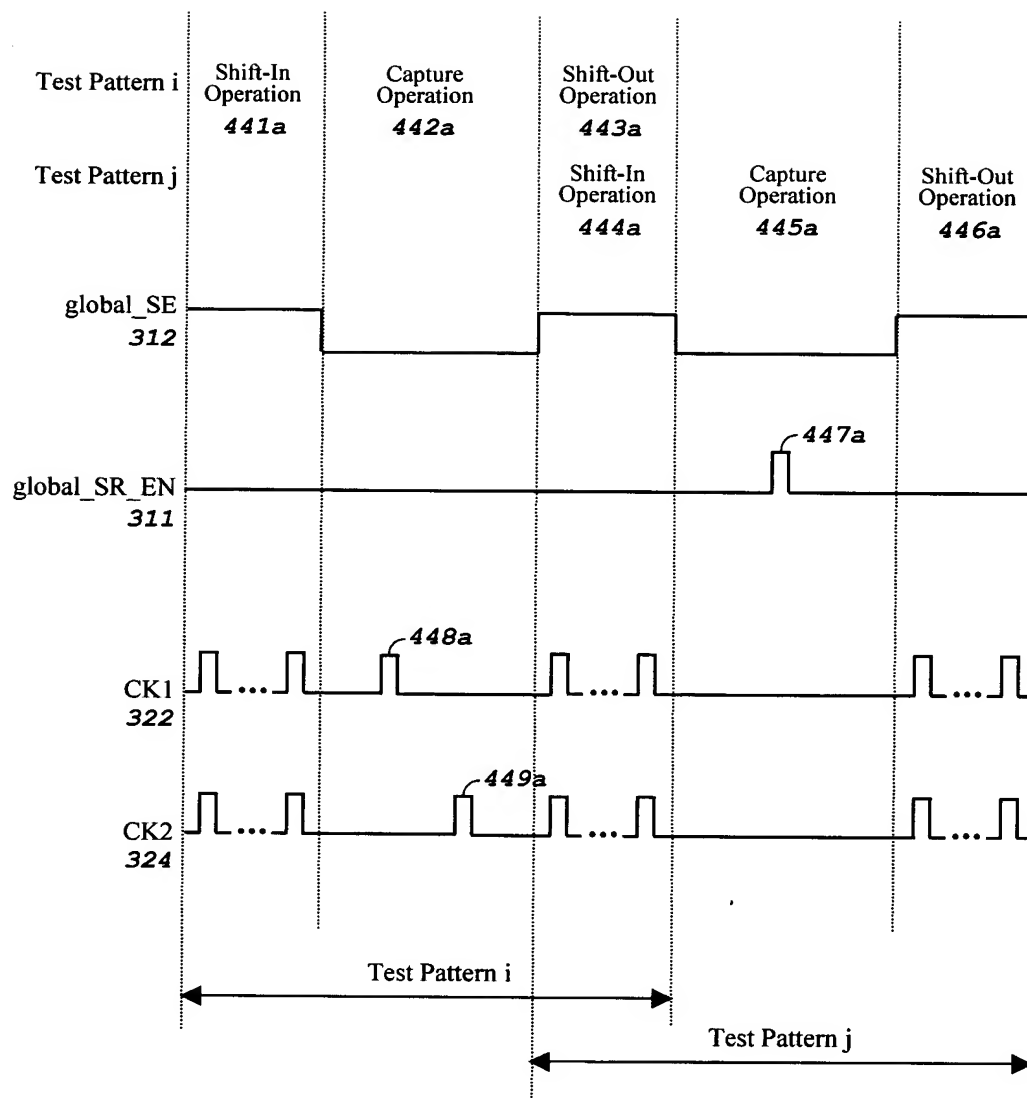


FIG. 4E

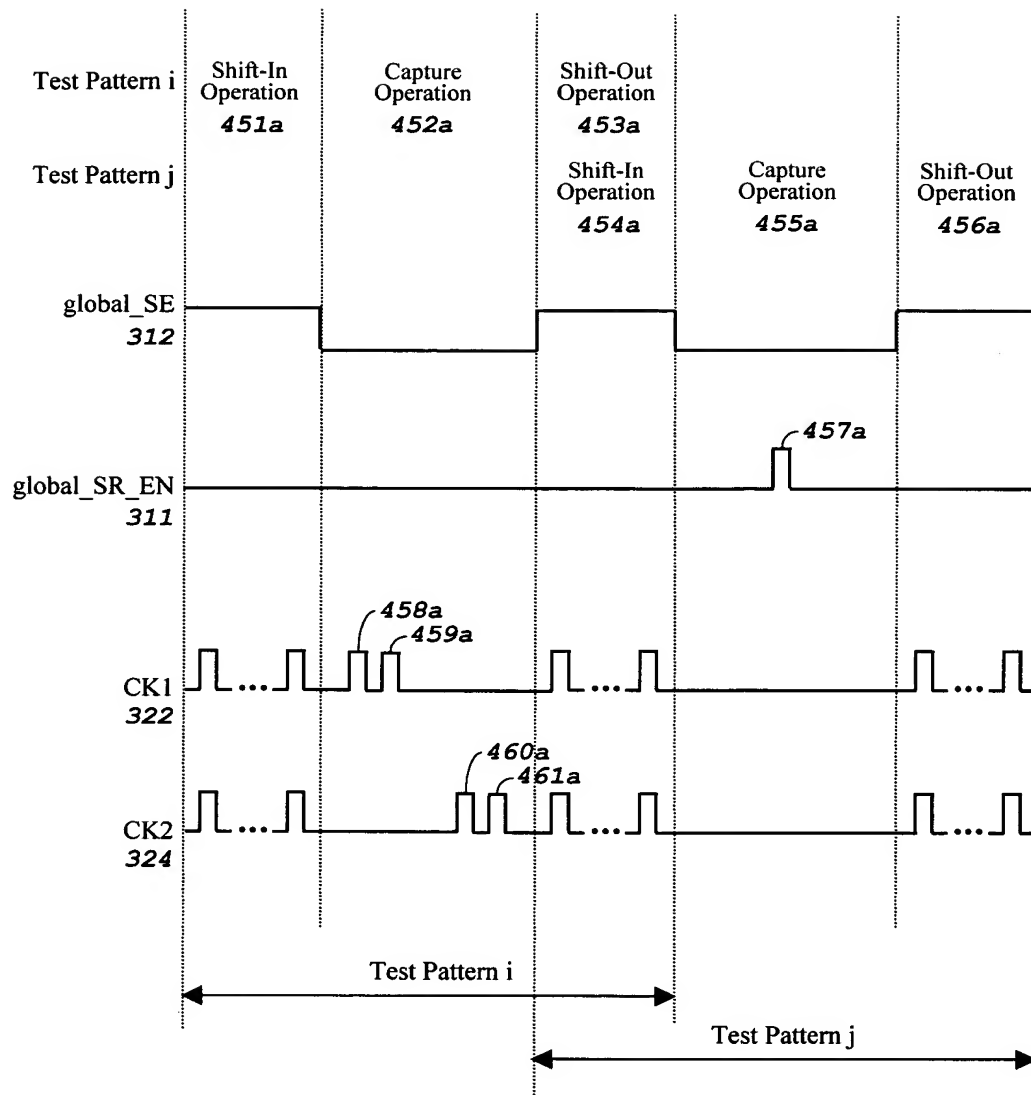


FIG. 4F



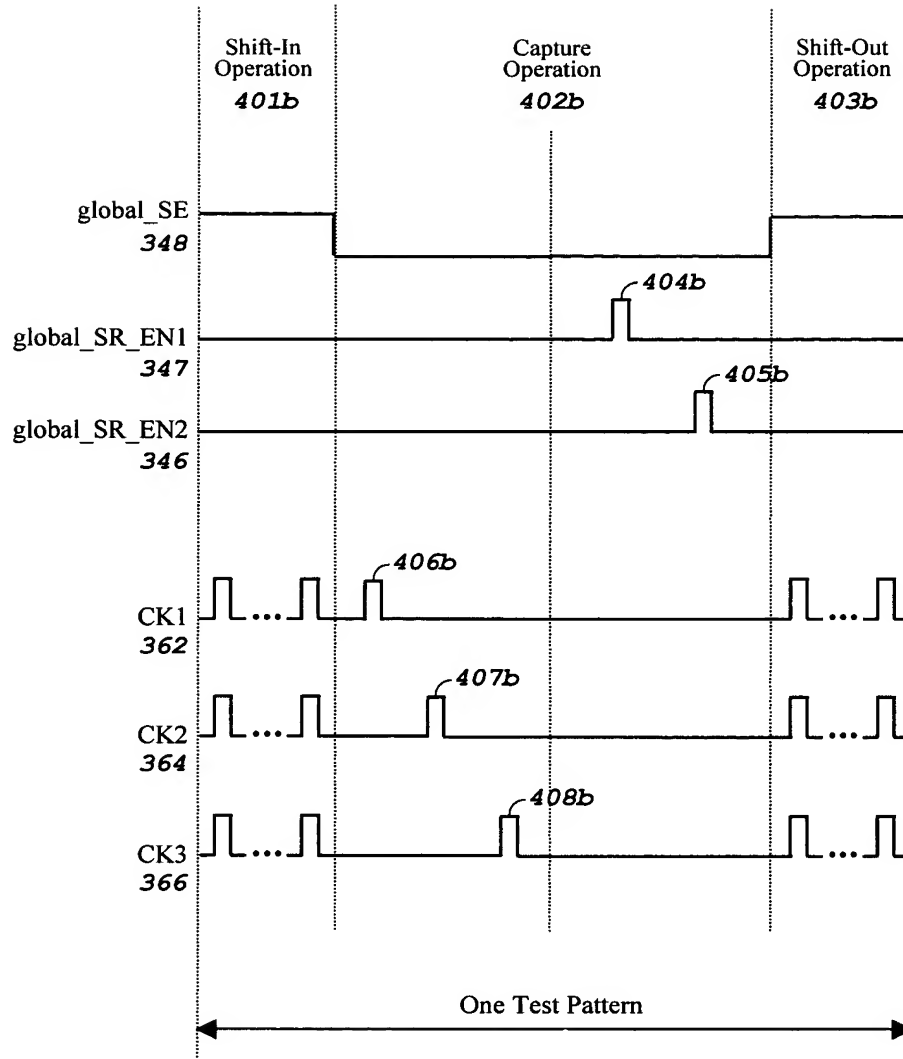


FIG. 4G

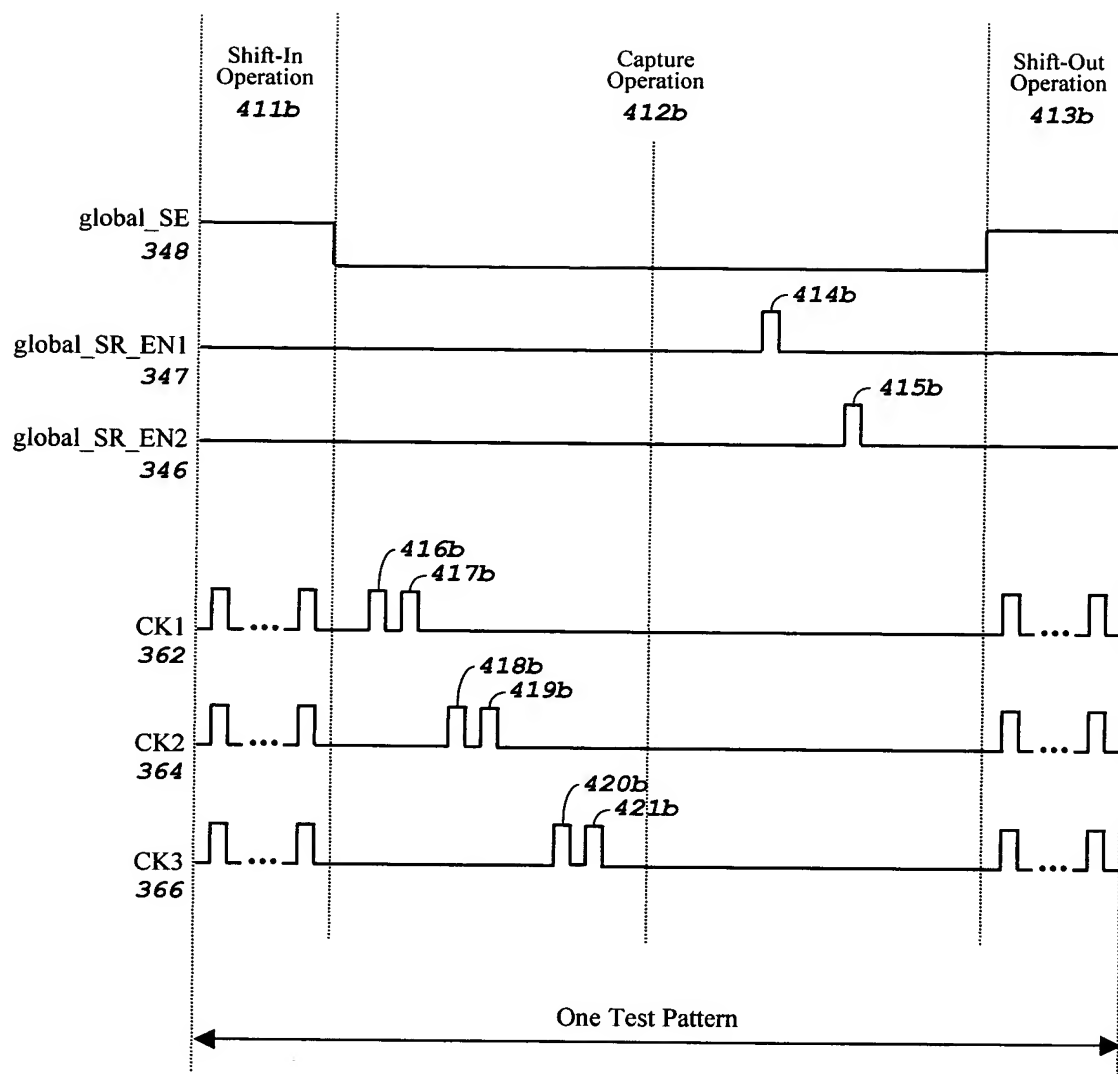


FIG. 4H

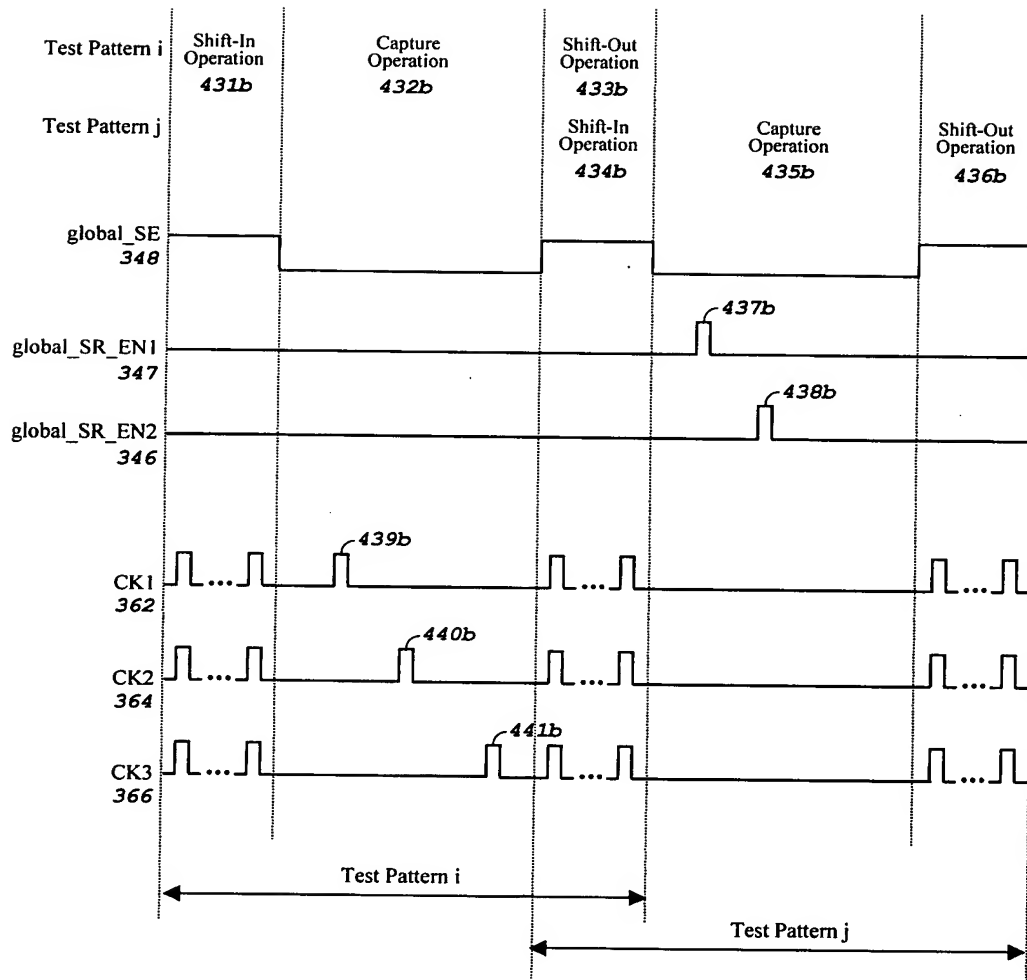


FIG. 4I

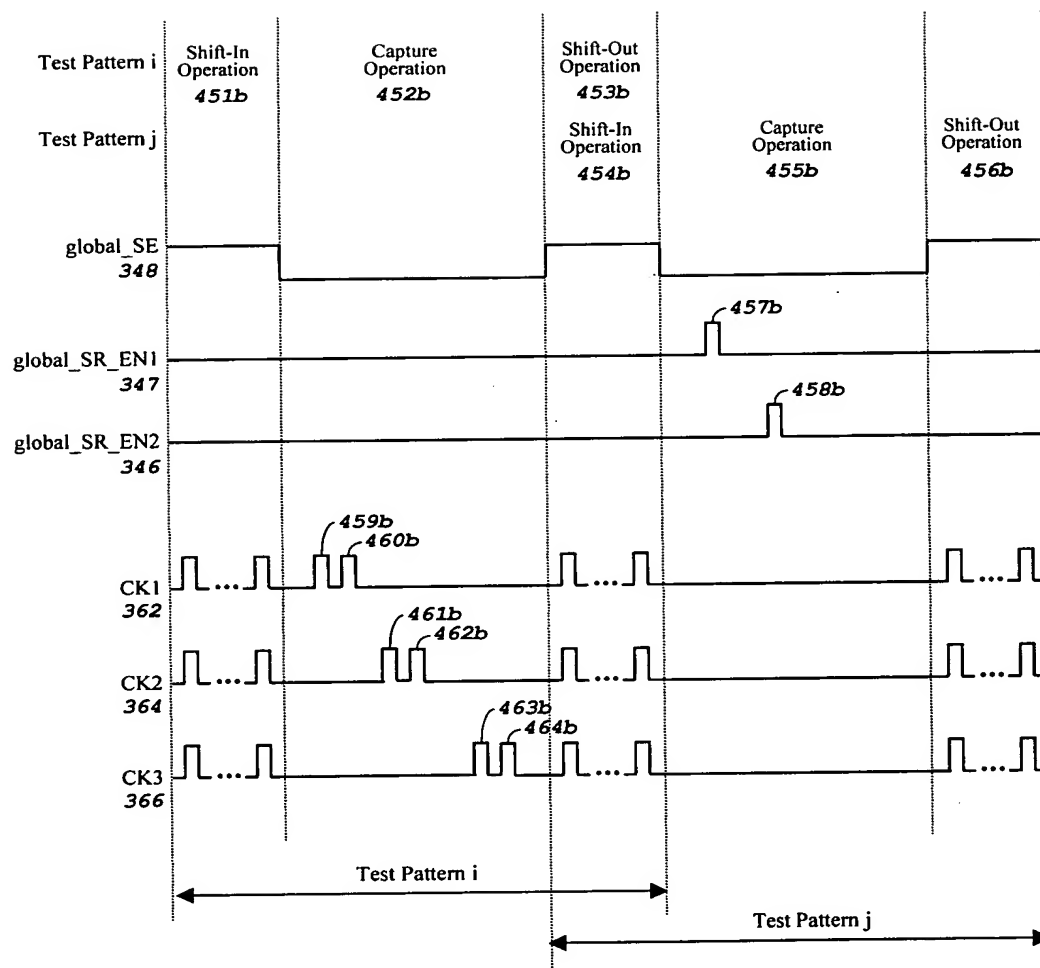


FIG. 4J

	Original	Modified
1	input rst, x, d, ck;	input rst, x, d, ck, SE, SR_EN;
2	wire s_rst, c_rst;	wire s_rst, c_rst;
3	reg z, q1, q2;	reg z, q1, q2;
4		
5	assign s_rst = rst & z;	// Added for repair.
6	assign c_rst = rst & x;	wire scan_s_rst;
7	always @(posedge ck)	wire scan_c_rst;
8	z <= x;	
9		
10	// s_rst is sequentially-gated reset.	// Added to repair sequentially-gated reset.
11	always @(posedge ck or posedge s_rst)	assign scan_s_rst = ~(SE   ~SR_EN) & s_rst;
12	if (s_rst)	// Added to repair combinational gated reset.
13	q1 <= 0;	assign scan_c_rst = ~(SE   ~SR_EN) & c_rst;
14	else	
15	q1 <= d;	assign s_rst = rst & z;
16		assign c_rst = rst & x;
17	// c_rst is combinational-gated reset.	always @(posedge ck)
18	always @ (posedge ck or posedge c_rst)	z <= x;
19	if (c_rst)	
20	q2 <= 0;	// Modified to repair sequentially-gated reset.
21	else	always @(posedge ck or posedge scan_s_rst)
22	q2 <= d;	if (scan_s_rst)
23		q1 <= 0;
24		else
25		q1 <= d;
26		
27		// Modified to repair combinational-gated reset.
28		always @(posedge ck or posedge scan_c_rst)
29		if (scan_c_rst)
30		q2 <= 0;
31		else
		q2 <= d;

FIG. 5A

	Original	Modified
1	input x, d, ck;	input x, d, ck, SE, SR_EN;
2	wire d_rst;	wire d_rst;
3	reg g_rst, q1, q2;	reg g_rst, q1, q2;
4		
5	assign d_rst = 1;	// Added for repair.
6	always @(posedge ck)	wire scan_g_rst;
7	g_rst <= x;	wire scan_d_rst;
8		
9	// g_rst is generated reset.	// Added to repair generated reset.
10	always @(posedge ck or posedge g_rst)	assign scan_g_rst = ~(SE   ~SR_EN) & g_rst;
11	if (g_rst)	// Added to repair destructive reset.
12	q1 <= 0;	assign scan_d_rst = ~(SE   ~SR_EN) & d_rst;
13	else	
14	q1 <= d;	
15		assign d_rst = 1;
16	// d_rst is destructive reset.	always @(posedge ck)
17	always @ (posedge ck or posedge d_rst)	g_rst <= x;
18	if (d_rst)	
19	q2 <= 0;	// Modified to repair generated reset.
20	else	always @(posedge ck or posedge scan_g_rst)
21	q2 <= d;	if (scan_g_rst)
22		q1 <= 0;
23		else
24		q1 <= d;
25		
26		// Modified to repair destructive reset.
27		always @(posedge ck or posedge scan_d_rst)
28		if (scan_d_rst)
29		q2 <= 0;
30		else
		q2 <= d;

FIG. 5B

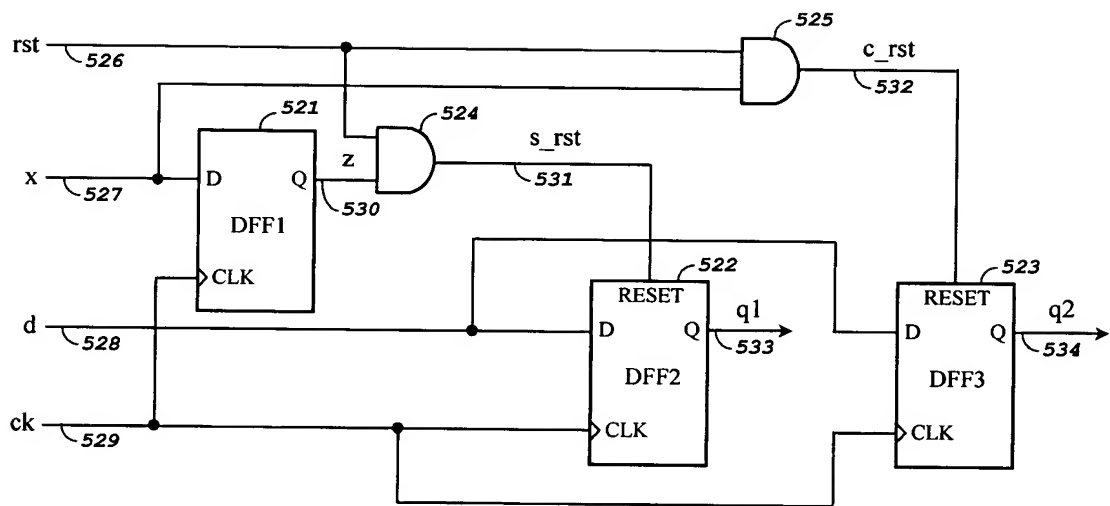


FIG. 5C

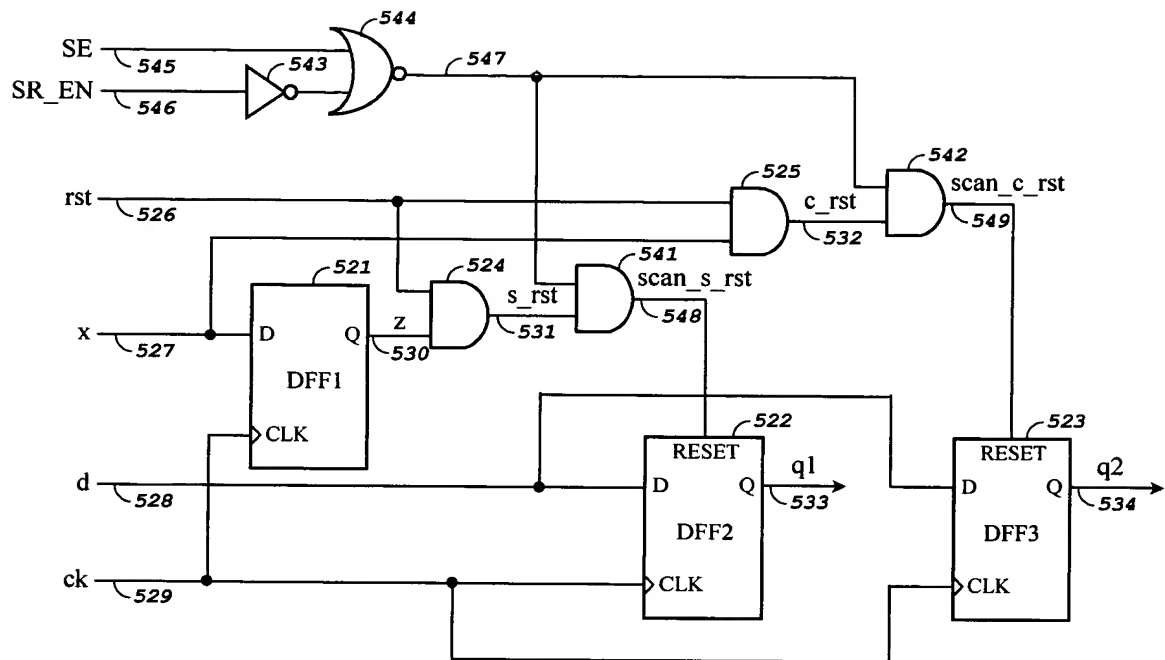


FIG. 5D



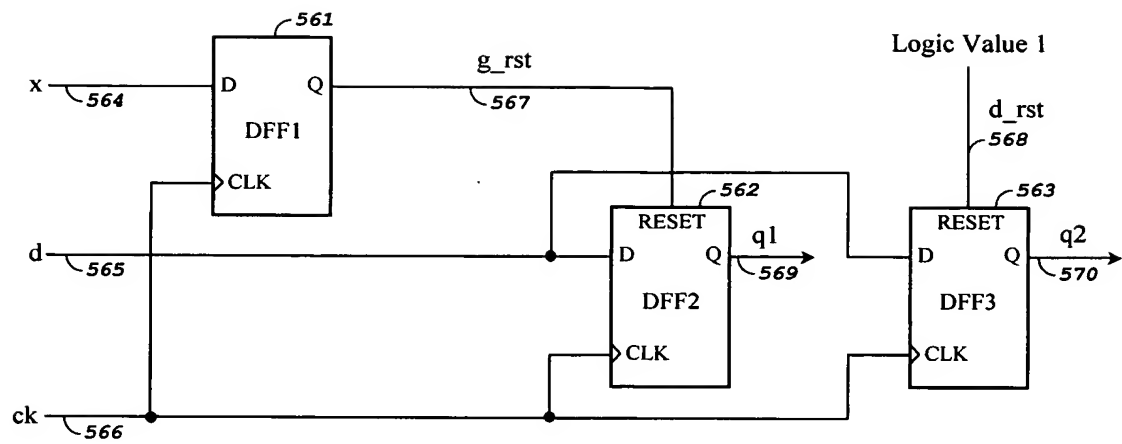


FIG. 5E

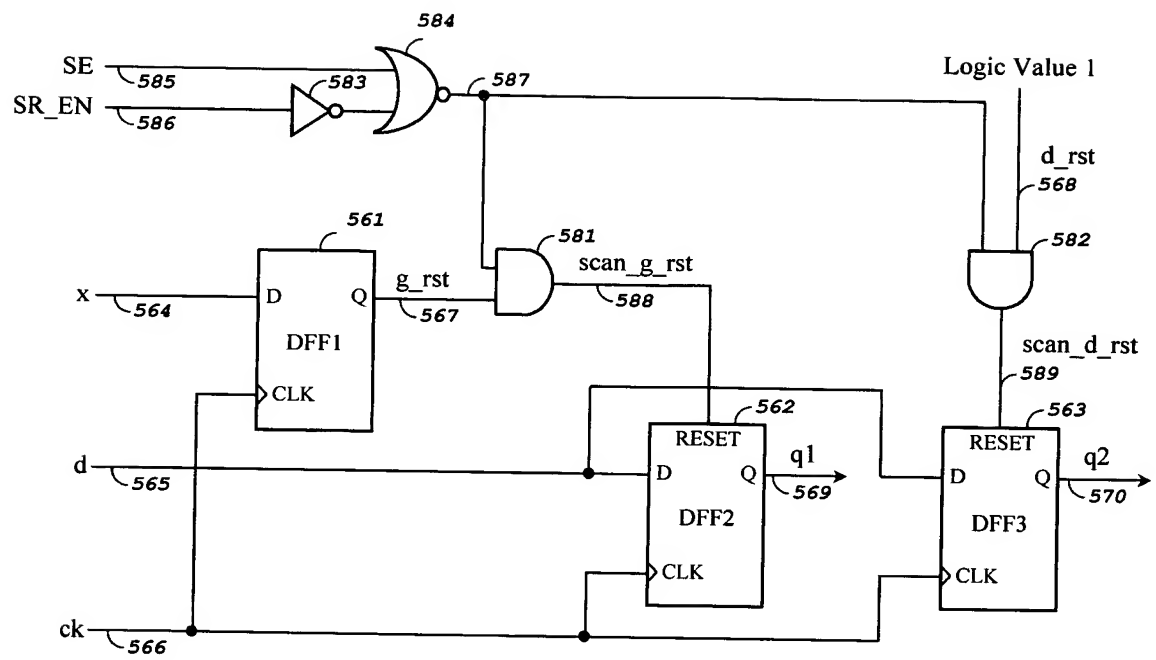


FIG. 5F

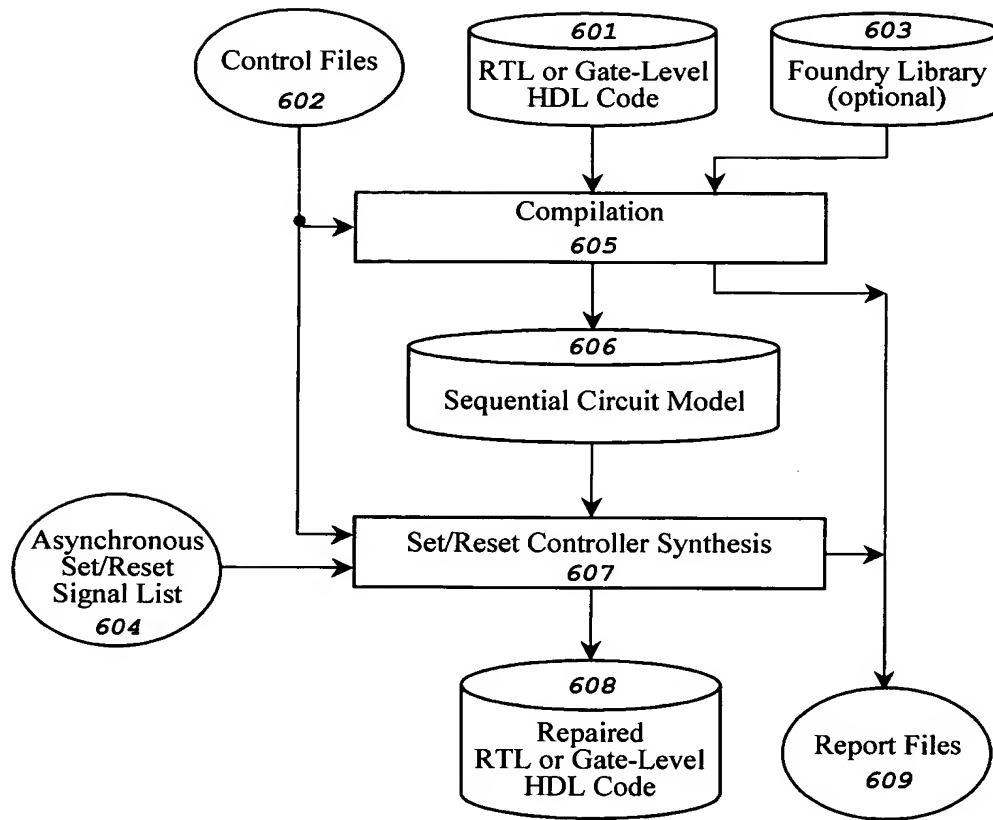


FIG. 6

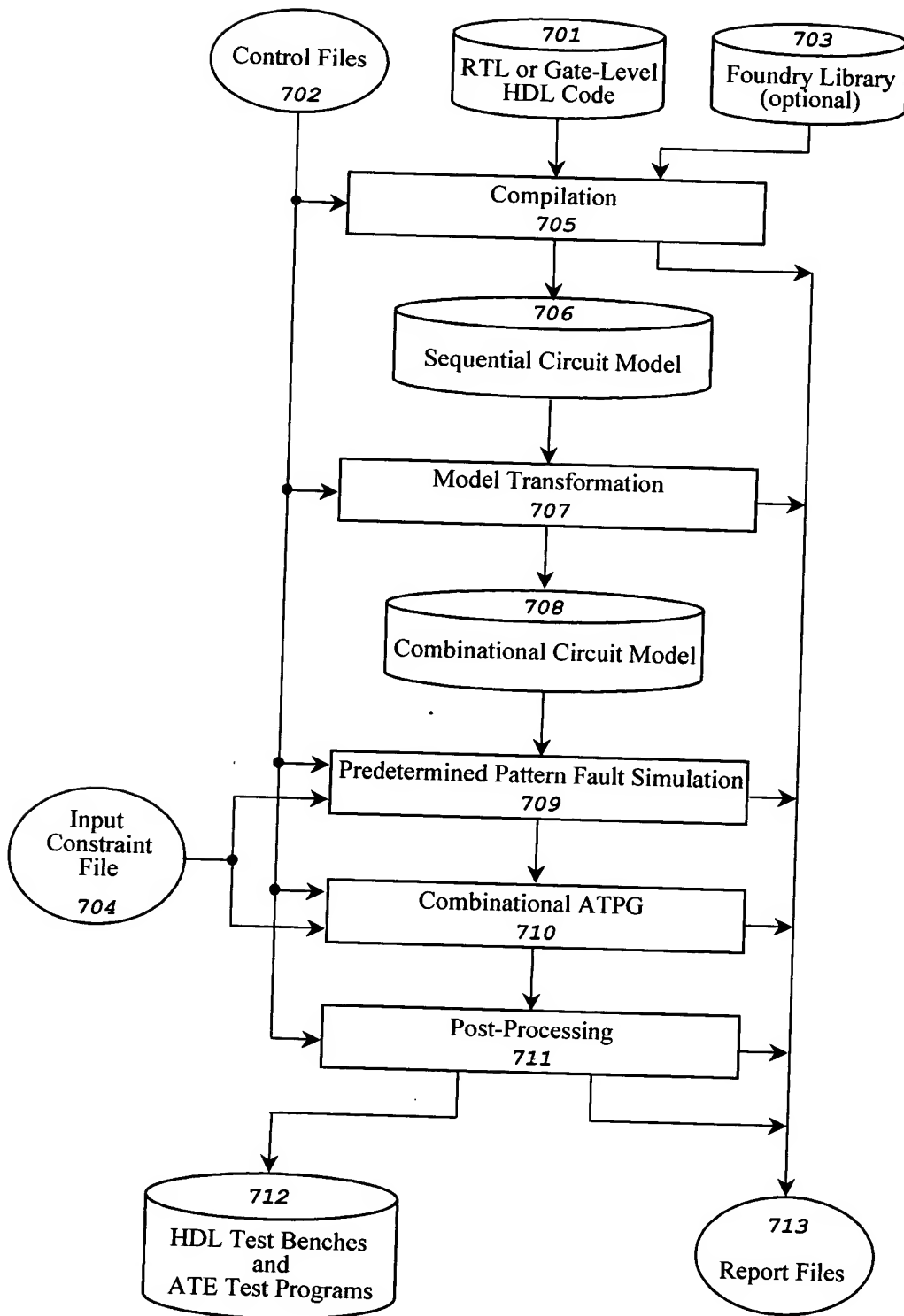


FIG. 7A

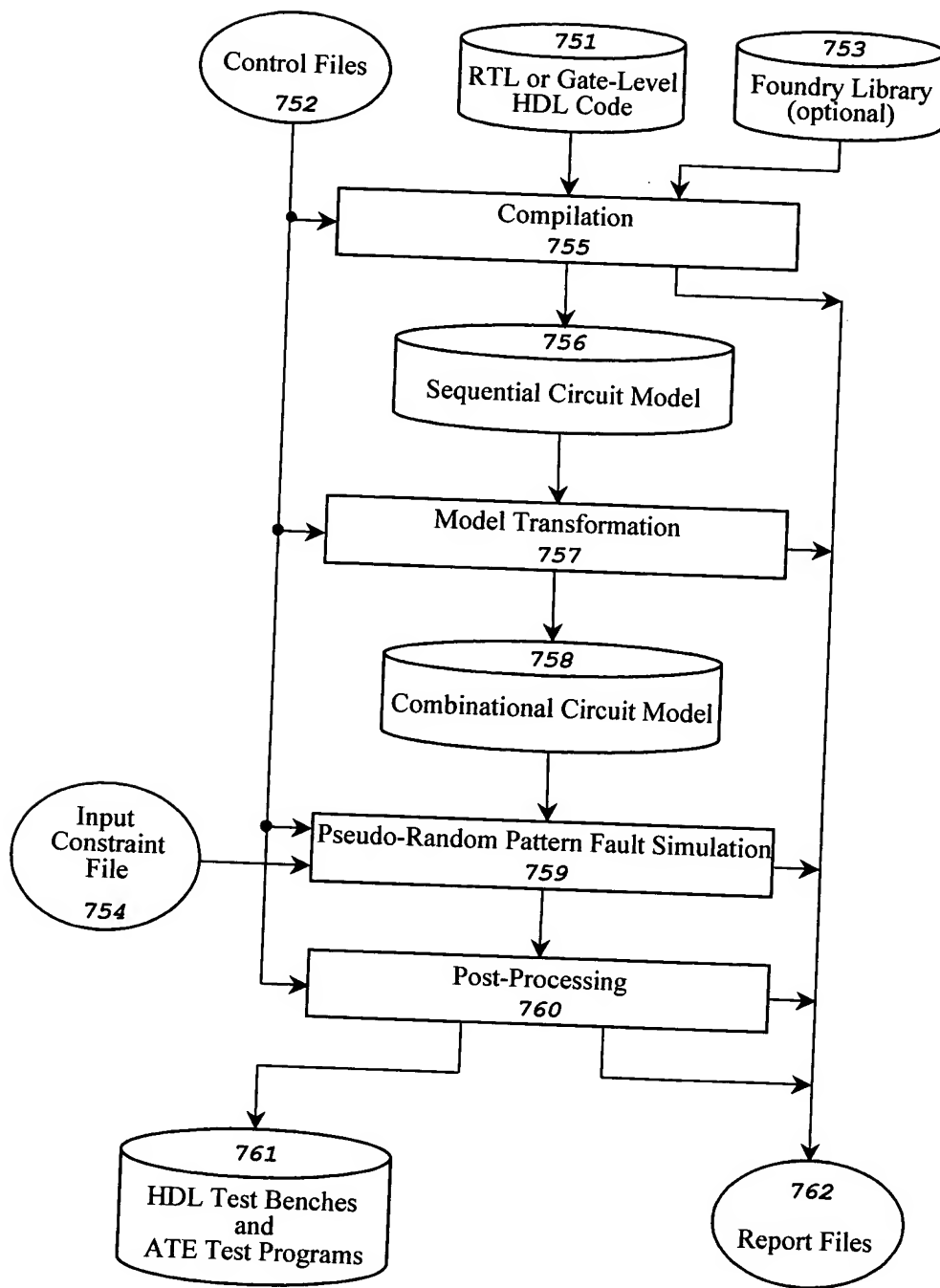


FIG. 7B

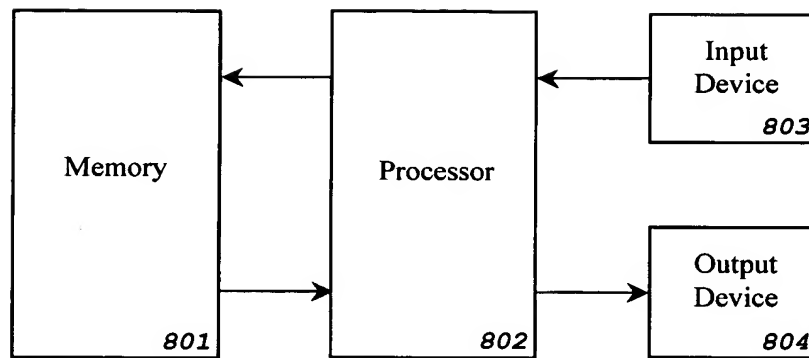


FIG. 8